

Date=09/09/2020

Lecture By=Arkesh Jaiswal

Subject ⇒ System Design=2

IN PREVIOUS LECTURE (QUICK RECAP) Date-08/02/2020	In Today's Lecture (Overview)
<a href="#">What is System Design</a> <a href="#">Inputs to System Design</a> <a href="#">Outputs for System Design</a> <a href="#">Types of System Design</a>  <a href="#">Mcqs</a>  <a href="#">Questions for the self practice</a>	<a href="#">Youtube Data Structure(How youtube works)</a>  <a href="#">Whatsapp data structure</a>  <a href="#">Mcqs</a>  <a href="#">Questions for self practice</a>

In today's lecture we learned about how Youtube And Whatsapp Datastructures work  
First let's Talk About youtube data structure

## Youtube Data Structure(How youtube works)

First thing is Api Design

### 1.API DESIGN

In today's world, a lot of systems support mobile platform so APIs are the best choices to be able to provide the distinction between developers and support mobile support as well. We can use REST or SOAP. A lot of huge companies prefer to REST or SOAP according to their systems. There are three main API's we will mention below:

*1- UploadVideo(apiKey, title, description, categoryID, language)*

Upload video is the first API that we should mention. There are basically five main properties of this API. You can add more properties to the UploadVideo API. Note that

apiKey is the developer key of the registered account of service. Thanks to apiKey we can eliminate hacker attacks. UploadVideo returns the HTTP response that demonstrates whether the video was uploaded successfully or not.

### *2- DeleteVideo (apiKey, videoID)*

Check if the user has permission to delete video. It will return HTTP response 200 (OK), 202 (Accepted) if the action has been queued, or 204 (No Content) based on your response.

### *3- GetVideo (apiKey, query, videoCountToReturn, pageNumber)*

Return JSON containing information about the list of videos and channels. Each video resource will have a title, creation date, like count, total view count, owner and other meta information.

\*\*There are more APIs to design video sharing services, however, these three APIs are more important than the others. Other APIs will be likeVideo, addComment, search, recommendation etc...

Second thing is Database Design

## **2.DATABASE DESIGN**

There are two choices to define the database schema. These are SQL and NoSQL. We can use traditional database management systems like MsSQL or MySQL to keep data. As you know, we should keep information about videos and users into RDBMS. Other information about videos, called metadata, should be kept too. Now we have the main

three tables to keep data. (Notice that we just only think of the basic properties of Youtube. We can forget the recommendation system).

### **For User**

- UserID
- Name
- Age
- Email Address
- Register Date (DateTime)
- Last Login (DateTime)

### **Video**

- VideoID (primary key – generated by KGS – Key generation service)
- VideoTitle (nvarchar)
- Size (float)
- UserID (foreign key with User Table)
- Description (nvarchar)
- CategoryID (int) : Note that we can create Category Table to define categories
- Number of Likes (int)

- Number of Dislikes (int)
- Number of Displayed (int) – We can use big int to keep displayed number
- Uploaded Date (DateTime)

### **VideoComment**

- CommentID (primary key)
- UserID (foreign key with User Table)
- VideoID (foreign key with Video Table)
- Comment (nvarchar)
- CommentDate (DateTime)

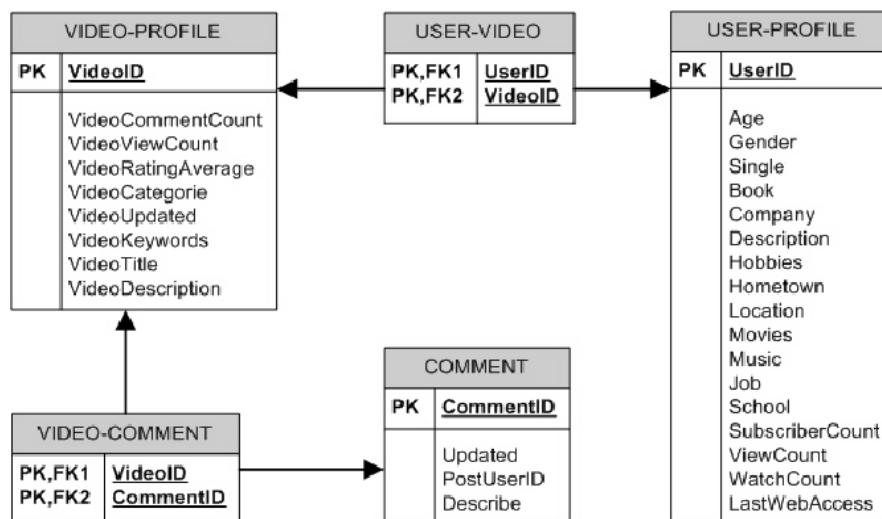
### **SYSTEM DESIGN CONSIDERATION**

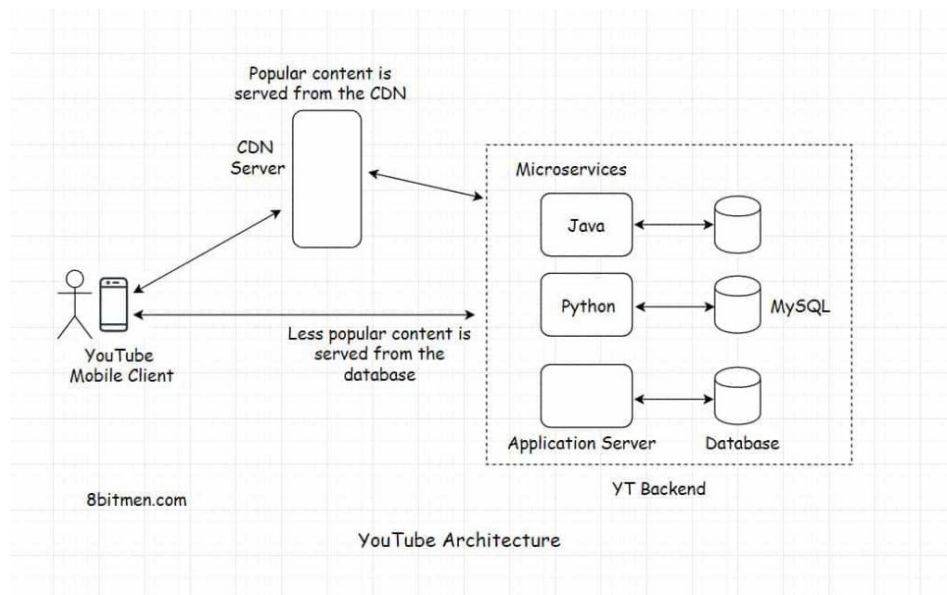
There are basic features found in web-based systems. The main ones are client, web server, application server, database and cache systems. Depending on the intensity of system traffic, the number of servers or services increases and the load balancer distributes incoming requests between these servers or services. Additionally, queues can be used depending on the density of incoming requesters. Queue operation helps users to keep from waiting more time to get respond. In our Youtube service;

- Client
- Web Server
- Application Server

- Database
- Video Storage
- Encode Service
- Queue
- Replication
- Redundancy
- Load balancing
- Sharding

We can distribute services to three parts to decrease response time because video uploading takes more time from video downloading. Video can be downloaded from the cache and getting data from the cache is a fast way. The client is basically users who use the system. Web Server is the first entity that meets the request. Incoming requests can take place in upload service, search service or download service.





## Whatsapp data structure

Field name	Meaning
_id	Sequence number of the record (set by SQLite)
key_remote_jid	WhatsApp ID of the communication partner
key_from_me	Message direction: '0'=incoming, '1'=outgoing
key_id	Unique message identifier
status	Message status: '0'=received, '4'=waiting on the
server, '5'=received at the destination,	

'6'=control message, '13'=message opened by the recipient (read)	
need_push	'2' if broadcast message, '0' otherwise
data	Message content when media_wa_type = '0'
timestamp	Contains a timestamp in the Unix Epoch Time (ms) format, the value is taken from the device clock
media_url	URL of the transmitted file (when media_wa_type={'1','2','3'})

media_mime_type	MIME type of the transmitted file (when media_wa_type={'1','2','3'})
media_wa_type	Message type: '0'=text, '1'=image, '2'=audio, '3'=video, '4'=contact card, '5'=geo position)
media_size	Size of the transmitted file (when media_wa_type={'1','2','3'})
media_name	Name of the transmitted file (when media_wa_type={'1','2','3'})
media_caption	Contains the words "audio" and "video" for corresponding values of media_wa_type (when media_wa_type={'1','3'})
media_hash	Base64-encoded SHA-256 hash of the transmitted file (when



receipt\_server\_timestamp

Time of receipt of the central server ACK.  
Contains a timestamp in the Unix Epoch Time (ms) format. The value is taken from the device clock (if key\_from\_me='1', '-1' otherwise)

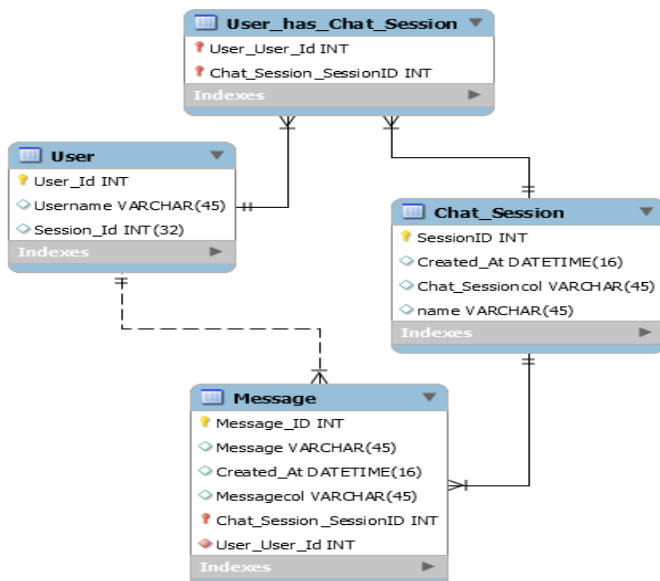
receipt\_device\_timestamp

Time of receipt of the recipient ACK.  
Contains a timestamp in the Unix Epoch Time (ms) format. The value is taken from the device clock (if key\_from\_me='1', '-1' otherwise)

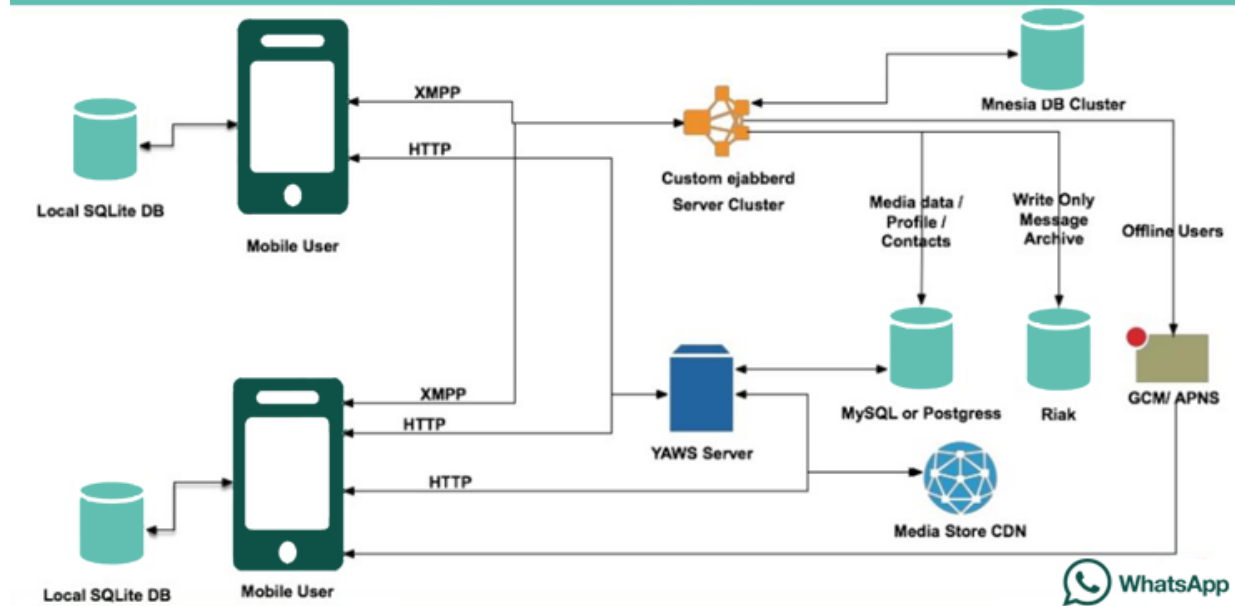
read\_device\_timestamp

Time of opening (reading) the message.  
Contains a timestamp in the Unix Epoch Time (ms) format. The value is taken from the device clock

media_duration	Duration in sec. of the transmitted media file (when media_wa_type={ '1','2','3' })
origin	'2' if broadcast message, '0' otherwise
latitude	Latitude of the message sender (when media_wa_type='5')
longitude	Longitude of the message sender (when media_wa_type='5')
thumb_image	Housekeeping information
remote_resource	ID of the sender (only for group chat messages)



# Understanding WhatsApp architecture



## Mcqs

1.What is the primary server (controller) of a web app?

A = Database

B = Cloud storage for static assets

C = BE logic

2.Where are sockets used?

A = Chat applications

B = Video sites

**3.Can MongoDB store entire image files inside objects/dictionaries**

A = Yes

B = No

**4. What is S3?**

A = Amazon's storage for static assets

B = Google's storage for static assets

## **Questions for self practice**

Assignment

1. Design Amazon e-commerce web app
2. Design a Library Management System